Towards a model-checker for counter systems

Stéphane Demri, Alain Finkel, Valentin Goranko, Govert van Drimmelen

Towards a model-checker for counter systems *

S. Demri[1], A. Finkel[1], V. Goranko[2], and G. van Drimmelen[2]

[1] LSV/CNRS UMR 8643 & INRIA Futurs projet SECSI & ENS Cachan
email: {demri,finkel}@lsv.ens-cachan.fr

[2] University of the Witwatersrand, Johannesburg
email: {govert,goranko}@maths.wits.ac.za

Abstract. This paper deals with model-checking of fragments and extensions of CTL* on infinite-state Presburger counter systems, where the states are vectors of integers and the transitions are determined by means of relations definable within Presburger arithmetic. We have identified a natural class of admissible counter systems (ACS) for which we show that the quantification over paths in CTL* can be simulated by quantification over tuples of natural numbers, eventually allowing translation of the whole Presburger-CTL* into Presburger arithmetic, thereby enabling effective model checking. We have provided evidence that our results are close to optimal with respect to the class of counter systems described above. Finally, we design a complete semi-algorithm to verify first-order LTL properties over trace-flattable counter systems, extending the previous underlying FAST semi-algorithm to verify reachability questions over flattable counter systems.

1 Introduction

Background Model-checking of infinite-state systems (for a survey see [BCMS01]) is a rapidly growing area of formal verification. It has been successfully applied to real-time and hybrid systems, concurrent systems, Petri nets, asynchronous communication devices (unbounded FIFO channels), infinite and unbounded data structures (counters, queues, lists), etc. The single most important property of practical interest in infinite-state transition systems is *state reachability* which is often undecidable in structures with otherwise decidable first-order theories, such as e.g., automatic structures. Therefore, intensive research has been devoted to identifying classes of finitely presentable infinite structures with decidable reachability and related safety properties.

Transition systems defined by Presburger relations provide a large natural class of infinite-state transition systems [BFLS05], suitable for modeling in various applications such as TTP Protocol (embedded system) [BFL04] and broadcast protocols [EFM99], to quote a few examples. Important cases of such transition systems with computable reachability have been established in [Iba78,FO97,CJ98,FL02]. The method of acceleration for computing reachability has been developed in [Boi98,FL02] and completely implemented in the verification tool FAST [BFL04].

* Supported by CNRS/NRF project No 15469.

Motivation For practical (computer-aided) model-checking, an infinite-state system must be provided with an effective finitary presentation, and in particular, must admit a symbolic representation of sets of states and transitions. *Presburger arithmetic* is a particularly appropriate platform for symbolic representation of a wide variety of infinite state systems, such as *counter systems* (see [BFLP03]) where vectors of integers are subjected to linear transformations from finite control graph. These strongly extend counter automata and even very simple examples of counter systems can have notoriously difficult and unpredictable behaviour, a witness being the Syracuse problem, see e.g. [Lag85]. An important and natural class of counter systems, in which various practical cases of infinite state systems (e.g. broadcast protocols [FL02]) can be modelled, are those with a *flat* control graph, i.e, those where no control state occurs in more than one simple cycle (see [Boi98,CJ98,CC00,FL02,BFLP03,Ler03]). Strong results on verifying safety and reachability properties on flat counter systems have been obtained in [CJ98,FL02]. However, so far such properties have not been considered in the framework of any formal specification language, and thus a natural question that arises is *to identify expressive logical languages in which formal specification and verification of properties of counter systems can be conducted.*

On the other hand, most of the studies on CTL*-model checking are restricted to (unfoldings of) finite transition systems, and few decidability results for CTL*-model checking on essentially infinite state systems are known [BEM97]. Actually, most of these results are immediate consequences of stronger results about decidable modal mu-calculus, or even the whole monadic second order logic in such systems, see e.g. [Wal01]. It is therefore important *to search for larger classes of effectively generated infinite state systems [without necessarily decidable MSO], but in which natural first-order extensions of CTL* have decidable model-checking.*

Our contribution We address jointly both problems described above, and we obtain a nearly optimal solution of them. Our main contributions are the following:

1. We introduce a Presburger extension of CTL*, where atomic propositions range over Presburger-definable sets of configuration states; we interpret that extension over Presburger counter systems, thus proposing a very powerful specification language for them. Presburger counter systems are understood as infinite-state transition systems with states being vectors of integers (counter values) and transition relations definable in Presburger arithmetic. This class of models naturally includes Minsky machines.
2. We identify a class of Presburger counter systems, on which local model checking problem for the Presburger-CTL* is decidable. These are Presburger counter systems defined over flat control graphs with arcs labelled by Presburger formulae for which counting acceleration over every cycle in the control graph is Presburger definable.
3. We show that the decidability result described above persists in a strong extension with a class of temporal operators defined by means of CQDD (see [BH99]) in a way analogous to Wolper's Extended temporal logic [Wol83].

4. We provide evidence that our results are close to optimal wrt the class of Presburger counter systems described above, by showing that small relaxations of each of the conditions lead to undecidability.
5. We design a complete semi-algorithm to check whether a given Presburger counter system satisfies a Presburger-LTL formula extending the underlying reachability semi-algorithm used in the tool FAST [BFL04].

Related work On the logical side, temporal logics with Presburger constraints have been developed in [BEH95,BGP97,CC00,SS04,BDR03], some of which have quite expressive decidable fragments. However, undecidability of the reachability problem can be proved for quite restricted counter systems, see e.g. [Cor02] while at the same time very few classes of counter systems are decidable for CTL* (see e.g. [FWW97] for one-counter systems). A logical formalism closer to the one developed in this paper is presented in [BGP97] where an undecidable temporal logic with CTL-like operators and atomic formulae in Presburger arithmetic is introduced and the models are counter systems. Model checking discrete timed automata with parametric timed CTL is also shown decidable by translation into Presburger arithmetic in [BDR03].

2 Preliminaries

Flat graphs. A directed labelled graph $\mathcal{G} = \langle \Sigma, Q, E \rangle$ is a structure such that Q is a non-empty set, Σ is a non-empty finite alphabet and $E \subseteq Q \times \Sigma \times Q$. As usual, $\langle q, a, q' \rangle \in E$ is also denoted by $q \xrightarrow{a} q'$. A **cycle** in a directed labelled graph is a closed path (where the initial and final vertices coincide) with no repeating edges. A **simple cycle** is a cycle in which the only repeated vertex is the initial (and final) vertex. We define the **length of a path** $\lambda = q_0 \xrightarrow{\psi_0} q_1 \ldots \xrightarrow{\psi_{n-1}} q_n$ (each $q_i \in Q$, $\psi_i \in \Sigma$), denoted $|\lambda|$, as n. A graph is **flat** if every cycle in it is a simple cycle. Graphs with a singleton alphabet are the standard directed graphs.

Presburger arithmetic. This is the first-order theory of the structure $\langle \mathbb{N}, + \rangle$. Given a Presburger formula $\psi(x_1, \ldots, x_n)$ with free variables in $\mathbf{x} = \langle x_1, \ldots, x_n \rangle$, and $\mathbf{a} = \langle a_1, \ldots, a_n \rangle \in \mathbb{N}^n$, the truth of $\psi(x_1, \ldots, x_n)$ with respect to the interpretation $\mathbf{a}$ is denoted by $\mathbf{a} \models \psi(\mathbf{x})$. Elements of $\mathbb{N}^n$ will be usually denoted by $\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$, ... and vectors of variables will be denoted by $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$, $\mathbf{t}$, ... (possibly decorated). A set $X \subseteq \mathbb{N}^n$ is said to be **Presburger definable** iff there is a Presburger formula $\psi(\mathbf{x})$ with free variables $\mathbf{x} = \langle x_1, \ldots, x_n \rangle$ such that $X = \{\mathbf{a} \in \mathbb{N}^n : \mathbf{a} \models \psi(\mathbf{x})\}$. A **binary relation of dimension** $n > 0$ is a relation $R \subseteq \mathbb{N}^n \times \mathbb{N}^n$; thus R is Presburger definable iff there is a Presburger formula $\psi(\mathbf{x}, \mathbf{x}')$ with free variables $\mathbf{x} = \langle x_1, \ldots, x_n \rangle$ and $\mathbf{x}' = \langle x'_1, \ldots, x'_n \rangle$ such that $R = \{\langle \mathbf{a}, \mathbf{b} \rangle \in \mathbb{N}^n \times \mathbb{N}^n : \mathbf{a}, \mathbf{b} \models \psi(\mathbf{x}, \mathbf{x}')\}$. Presburger arithmetic is known to be decidable and therefore, all the problems in the forthcoming sections that can be reduced to Presburger arithmetic are decidable.

Definition 1. *Let f be a partial function from $\mathbb{N}^n$ to $\mathbb{N}^n$ whose domain is* $\mathrm{dom}(f)$.

- *f is a **translation** if there exists $\mathbf{b} \in \mathbb{Z}^n$ such that for every $\mathbf{a} \in \mathrm{dom}(f)$ we have $f(\mathbf{a}) = \mathbf{a} + \mathbf{b}$.*
- *f is **linear** if if there exist a matrix $A \in \mathbb{N}^{n \times n}$ and $\mathbf{b} \in \mathbb{Z}^n$ such that for every $\mathbf{a} \in \mathrm{dom}(f)$ we have $f(\mathbf{a}) = A\mathbf{a} + \mathbf{b}$.*
- *f is **piecewise-linear** if there exists a finite partition of the domain $\mathrm{dom}(f) = \bigcup_{i=1}^{k} D_i$ so that the restriction on each D_i is linear.*
- *f is **Presburger definable** iff the graph of f is a Presburger definable relation.*

3 Temporal Logics for Presburger Counter Systems

In this section, we introduce a Presburger variant of standard temporal logic CTL* interpreted over Presburger transition systems.

3.1 Presburger Counter Systems

The infinite-state systems for which we investigate model checking are finitely represented by Presburger counter systems.

Definition 2. *A **Presburger counter system (PCS) of dimension** n, $\mathcal{C} = \langle \Sigma, Q, T \rangle$, is a tuple consisting of a finite set of **control states** Q, a finite set Σ composed of Presburger formulae of the form $\psi(\mathbf{x}, \mathbf{x}')$ encoding binary Presburger relations of dimension n and a set of **control transitions** $T \subseteq Q \times \Sigma \times Q$.*

- *$\mathcal{C}$ is **functional** if every element in Σ defines a partial function.*
- *a functional PCS $\mathcal{C}$, is **linear** [resp. **piecewise-linear**] if every element in Σ defines a linear [resp. piecewise-linear] function.*
- *a functional PCS $\mathcal{C}$ is a **counter automaton** if every element in Σ defines a translation.*

A PCS is therefore a labelled graph with alphabet made of specific Presburger formulae. A PCS is **flat** if its underlying control graph is flat.

Proposition 1. *It is decidable whether a given PCS is functional, linear, or a counter automaton.*

Every PCS $\mathcal{C} = \langle \Sigma, Q, T \rangle$ of dimension n naturally induces a graph $\langle S_{\mathcal{C}}, \rightarrow_{\mathcal{C}} \rangle$ (called a Presburger transition system) such that $S_{\mathcal{C}} = Q \times \mathbb{N}^n$ (set of configurations) and $\langle q, \mathbf{a} \rangle \rightarrow_{\mathcal{C}} \langle q', \mathbf{a}' \rangle$ iff there is $\langle q, \psi(\mathbf{x}, \mathbf{x}'), q' \rangle \in T$ such that $\mathbf{a}, \mathbf{a}' \models \psi(\mathbf{x}, \mathbf{x}')$. Wlog, we can assume that $S_{\mathcal{C}}$ is a subset of $\mathbb{N}^{n+1}$. Depending on the context, the configurations of $S_{\mathcal{C}}$ are indifferently written as $\mathbf{a} \in \mathbb{N}^{n+1}$ (control state encoded in the first element of $\mathbf{a}$), $\langle q, \mathbf{a} \rangle \in Q \times \mathbb{N}^n$ or as $\langle q, \mathbf{a} \rangle \in Q \times \mathbb{N}^{n+1}$ (with redundancy). A **configuration path** in $\mathcal{C}$ is an infinite path in the Presburger transition system of $\mathcal{C}$.

3.2 A Presburger Temporal Logic FOCTL*(Pr)

We now define a version of first-order CTL* that is appropriate for reasoning about Presburger transition systems of Presburger counter systems. The logic FOCTL*(Pr) differs from standard CTL* in the definition of atomic formulae. Whereas propositional variables are used in the propositional CTL*, we will use Presburger predicates, interpreted on the set of configurations, as the atomic formulae in FOCTL*(Pr). We introduce a countable set of individual variables, say VAR $= \{y, z, t \ldots\}$, for quantification over counter values. Elements of VAR are distinct from the distinguished ones in $\{x_0, x_1, \ldots, x_n\}$ that are free variables interpreted by the values of counters on configurations (the control state is encoded by x_0). In order to match the dimension of the models where such formulae will be interpreted, the Presburger predicates must have a matching number of free variables, thus giving a family of logics FOCTL*(Pr)$[n]$ parameterised by the dimension $n \geq 1$. When the dimension n is clear from the context, we just refer to FOCTL*(Pr). Atomic formulae are Presburger formulae of the form $\psi(\mathbf{x}, \mathbf{y})$ where $\mathbf{x} = x_0, x_1, \ldots, x_n$ and $\mathbf{y}$ is a vector of variables from VAR.

Formulae of FOCTL*(Pr)$[n]$ are defined as follows:

$$\varphi ::= \psi(\mathbf{x}, \mathbf{y}) \mid \neg\phi \mid \varphi \wedge \varphi \mid \mathtt{X}\varphi \mid \varphi \mathtt{U} \varphi \mid \mathtt{A}\ \varphi \mid \exists\ y\ \varphi,$$

where $y \in$ VAR. We shall freely use standard abbreviations such as the existential quantifier $\mathtt{E}$, the always operator $\mathtt{G}$ and the sometimes operator $\mathtt{F}$. The LTL fragment of FOCTL*(Pr) is made of formulae of the form either $\mathtt{E}\ \phi'$ or $\mathtt{A}\ \phi'$ where ϕ' has no path quantifiers. We define the **strict EF fragment** of FOCTL*(Pr) as the set of FOCTL*(Pr) formulae containing only the temporal operator $\mathtt{E\ F}$ and no nested occurrences of $\mathtt{E\ F}$.

Let π be an infinite configuration path of the system. Denote by $\pi_{\leq i}$ the initial part of π up to and including position i. Denote by π_{i+} the suffix of π starting at position i. We will give semantics of FOCTL*(Pr) over Presburger transition systems. To avoid the technical complications arising from terminating paths, we will impose the additional assumption that every configuration has some successor. This requirement can be satisfied by adding additional 'idle' states and corresponding 'idle' transitions. The satisfaction relation $\models$ is parameterized by an *environment* ρ that is a map VAR $\to \mathbb{N}$ in order to interpret the free variables from VAR that occur in formulae (we omit it when irrelevant). For a PCS $\mathcal{C} = \langle \Sigma, Q, T \rangle$ with Presburger transition system $\langle S_\mathcal{C}, \to_\mathcal{C} \rangle$, the satisfaction relation $\models$ is defined at position i of configuration path π as follows:

- $\pi, i \models_\rho \psi(\mathbf{x}, \mathbf{y})$ iff $\pi(i), \rho \models \psi(\mathbf{x}, \mathbf{y})$ in Presburger arithmetic,
- $\pi, i \models \neg\varphi$ iff $\pi, i \not\models \varphi$, $\pi, i \models \varphi \vee \varphi'$ iff $\pi, i \models \varphi$ or $\pi, i \models \varphi'$,
- $\pi, i \models \mathtt{X}\varphi$ iff $\pi, i+1 \models \varphi$,
- $\pi, i \models \varphi \mathtt{U} \varphi'$ iff there is some $j \geq i$ such that $\pi, j \models \varphi'$ and for each k, if $i \leq k < j$ then $\pi, k \models \varphi$,
- $\pi, i \models \mathtt{A}\ \varphi$ iff for every infinite configuration path π' such that $\pi'_{\leq i} = \pi_{\leq i}$ we have $\pi', i \models \varphi$,
- $\pi, i \models_\rho \exists y \varphi$ iff there is $m \in \mathbb{N}$ such that $\pi, i \models_{\rho[y \leftarrow m]} \varphi$ where $\rho[y \leftarrow m]$ is the environment obtained from ρ by only forcing y to be interpreted by m.

Apart from standard temporal properties encoded in CTL* (like liveness for instance) here are a few interesting properties that can be expressed by adding quantification over counter values:
Determinism: The graph restricted to the set of configurations reachable from the initial one is deterministic: $\mathtt{A\ G} \bigwedge_{0\leq i\leq n} \neg\exists y(\mathtt{E\ X}(x_i = y) \wedge \mathtt{E\ X}(x_i \neq y))$.
Boundedness: The transition graph restricted to the set of configurations reachable from the initial configuration is finite: $\exists y \mathtt{A\ G} \bigwedge_{1\leq i\leq n} x_i \leq y$.

We define below our basic problems. In the local model-checking problem considered here, we assume that all variables of the FOCTL*(Pr)$[n]$ formula, except those in $\mathbf{x}$, are bound. We will call such formulae **semi-closed**. In that way, we do not need to specify an environment in the statement below.

1. LOCAL MODEL CHECKING: Given an PCS $\mathcal{C}$ with Presburger transition system $\langle S_\mathcal{C}, \rightarrow_\mathcal{C}\rangle$, a configuration $\langle q, \mathbf{a}\rangle \in S_\mathcal{C}$, and a FOCTL*(Pr)$[n]$ formula ϕ, determine if for every path π such that $\pi(0) = \langle q, \mathbf{a}\rangle$, we have $\pi, 0 \models \phi$ (noted $\mathcal{C}, \langle q, \mathbf{a}\rangle \models \phi$).
2. VALIDITY CHECKING WITH AN INITIAL CONDITION: Given a PCS $\mathcal{C}$ with Presburger transition system $\langle S_\mathcal{C}, \rightarrow_\mathcal{C}\rangle$, a Presburger formula $\psi_0(\mathbf{x})$ and a FOCTL*(Pr)$[n]$ formula ϕ, check whether for every configuration $\langle q, \mathbf{a}\rangle$ satisfying $\psi_0(\mathbf{x})$, for every configuration $\langle q', \mathbf{a}'\rangle$ reachable from $\langle q, \mathbf{a}\rangle$, we have $\mathcal{C}, \langle q', \mathbf{a}'\rangle \models \phi$.

Variants of these problems can be defined by considering subclasses of PCS or other specification languages.

4 Towards Verification of Flattable PCS

Local model checking of FOCTL*(Pr) over the whole class of PCS is known to be highly undecidable even though reachability can be decided for many classes of counter systems, see e.g. [ISD$^+$00,CJ98,FL02,DPK03]. In the tool FAST, such a problem is solved by enumerating flattenings of some initial PCS and checking whether there is a flattening with the same reachability set. Many systems arising from applications do not have the desired flatness property, but are equivalent (in terms of the reachability relation) to flat systems. Such *flattable* systems, studied in [LS05], include e.g., reversal-bounded counter automata [Iba78]. In this section, we go one step further and propose a notion of flattening that can preserve sets of traces.

4.1 PCS with Decidable Reachability

Apart from flatness, Presburger counting acceleration property defined below is a key property to handle model-checking of PCS with a rich specification language as FOCTL*(Pr).

Definition 3. *For relation $R \subseteq \mathbb{N}^n \times \mathbb{N}^n$ we define the* ***counting acceleration of*** *R, as a relation $R_{\mathbf{CA}} \subseteq \mathbb{N}^n \times \mathbb{N} \times \mathbb{N}^n$ such that $\langle \mathbf{a}, i, \mathbf{b}\rangle \in R_{\mathbf{CA}}$ iff $\langle \mathbf{a}, \mathbf{b}\rangle \in R^i$.*

*R **has a Presburger counting acceleration** if its counting acceleration is Presburger definable.*

The **cycle relation** R^λ of a cycle λ in a PCS is the composition of local transition relations of the transitions on the cycle. More formally, a cycle λ is a sequence $t_1, \ldots, t_\alpha$ of transitions of the form $t_i = q_i \xrightarrow{\psi_i} q'_i$ such that for $0 \leq i \leq \alpha - 1$, $q_{i+1} = q'_i$ and $q_1 = q'_\alpha$. We define the local relation R^{t_i} as the set of pairs $\{\langle\langle q_i, \mathbf{a}\rangle, \langle q'_i, \mathbf{a}'\rangle\rangle : \mathbf{a}, \mathbf{a}' \models \psi_i(\mathbf{x}, \mathbf{x}')\}$. The relation R^λ is then $R^{t_1} \circ \cdots \circ R^{t_\alpha}$ ($\alpha - 1$ compositions). A cycle **has the Presburger counting acceleration property** if its cycle relation has a Presburger counting acceleration.

Definition 4. *A PCS $\mathcal{C}$ **has the Presburger counting acceleration property** if every cycle in the control graph of $\mathcal{C}$ has that property.*

Observe that if a PCS $\mathcal{C}$ has the Presburger counting acceleration property, we can effectively compute the Presburger formula associated to each cycle. It is sufficient to enumerate Presburger formulae $\psi(\mathbf{x}, i, \mathbf{y})$ and test whether

$$\forall \mathbf{x}, \mathbf{x}' \ (\psi(\mathbf{x}, 0, \mathbf{x}') \Leftrightarrow (\mathbf{x} = \mathbf{x}')) \wedge (\forall\, \mathbf{x}, \mathbf{x}', i \ \ \psi(\mathbf{x}, i{+}1, \mathbf{x}') \Leftrightarrow (\exists \mathbf{x}'' \ \psi(\mathbf{x}, i, \mathbf{x}'') \wedge \psi'(\mathbf{x}'', \mathbf{x}')))$$

is valid, where $\psi'(\mathbf{x}, \mathbf{y})$ is the effect of a given cycle. This is an instance of a more general result from [Ler06]. We also know that there exist counter systems of dimension 1 that do not have the Presburger counting acceleration property (for instance, consider the update $x'_1 = 2x_1$). In general, we expect that determining whether a counter system has a Presburger counting acceleration is an undecidable problem by extending similar results from [Ler06].

Flatness is another key property for PCS. For instance, every flat and linear PCS with the finite monoid property has the Presburger counting acceleration property [FL02] where a linear PCS has the finite monoid property if for every cycle λ in the system, the multiplicative monoid generated by the matrix of the linear function defining R^λ is finite (linear functions are closed under composition). Consequently, the Presburger formula defining the reachability relation in every flat and linear PCS with the finite monoid property is effectively computable. This consequence is incomparable with the main result from [CJ98]. Indeed, flatness is assumed in [CJ98] but not the finiteness of the monoid. Moreover [CJ98] and [FL02] have different and incomparable Presburger formulae labelling the transitions. For instance, transition relations in [CJ98] are not necessarily functional but they are restricted to relations on two variables. In Definition 5, the systems are more general than the ones in [CJ98] since we allow richer Presburger transition formulae.

Here we identify a large and natural class of Presburger counter systems for which model-checking of CTL* is decidable in addition to reachability.

Definition 5. *An **admissible Presburger counter system (ACS)** is a flat, functional PCS, that has the Presburger counting acceleration property.*

In particular, every flat and linear PCS with the finite monoid property is admissible. As observed in [FL02], flatness is the key property to be able to compute the reachability relation.

Proposition 2. *For every flat PCS satisfying the Presburger counting acceleration property (including ACS), one can effectively compute the reachability relation $\rightarrow_{\mathcal{C}}^{*}$ for $\langle S_{\mathcal{C}}, \rightarrow_{\mathcal{C}} \rangle$.*

The proof of Proposition 2 is based on the fact that essentially there is a finite number of types of configuration paths (see details later on) and one can effectively compute Presburger formulae associated to cycles. Definition 5 is close to optimal because relaxing any of the conditions for admissibility could easily lead to undecidability of the reachability problem, as indicated below.

Proposition 3. *The reachability problem is not decidable for all: (1) flat linear PCSs [Cor02], (2) linear PCSs with the finite monoid property (even counter automata) [Min67] and (3) flat piecewise-linear PCSs with a single control state and control transition [Min67].*

As a matter of fact, any counter automaton can be encoded as a flat piecewise-linear PCS with a single control state q_0 and control transition. Indeed, suppose that $q \xrightarrow{x:=x+1} q'$ is a transition in the counter automaton with the integer n [resp. n'] attached to q [resp. q'], then in the piecewise-linear PCS the unique transition is of the form $q_0 \xrightarrow{(x_0=n \wedge x'_0=n' \wedge x'=x+1) \vee \ldots} q_0$. There is an obvious correpondence between the transitions in the original counter automaton and the number of disjuncts in the Presburger formula labelling the unique transition.

4.2 Model-Checking for Three Main Classes of Flattable Systems

We establish in Section 5 that ACS have numerous desirable properties. For instance, FOCTL*(Pr) local model checking is decidable. However, it should not come as a surprise that the class of ACS forms a quite restricted subclass of PCS and numerous abstractions of communication protocols, concurrent systems and the like are not exactly ACS. More interestingly, many questions on specific classes of PCS can be reduced in a systematic way to reachability questions on ACS, see e.g. [FO97,CJ98,BFLP03] and a more thorough study in [LS05]. In this section, we provide the basis to understand how our results on ACS can be used to verify more general classes of PCS and under which hypotheses (see also Section 5.2). The most standard way to reduce a PCS to an ACS is via a graph homomorphism, aka a flattening [BFLS05].

Definition 6. *Let $\mathcal{C} = \langle \Sigma, Q, T \rangle$ and $\mathcal{C}' = \langle \Sigma', Q', T' \rangle$ be PCS of the same dimension and f be a function $f : Q' \rightarrow Q$. $\mathcal{C}'$ is a f-**flattening** of $\mathcal{C}$ iff $\mathcal{C}'$ is flat, $\Sigma' \subseteq \Sigma$, for every $\langle q, \psi(\mathbf{x}, \mathbf{x}'), q' \rangle \in T'$, we have $\langle f(q), \psi(\mathbf{x}, \mathbf{x}'), f(q') \rangle \in T$.*

When $\mathcal{C}'$ is a f-**flattening** of $\mathcal{C}$, $\mathcal{C}$ can be viewed as an abstraction of $\mathcal{C}'$.

The tool FAST [BFL04] generates flattenings via an exhaustive search algorithm. However, verification of FOCTL*(Pr) properties of $\mathcal{C}$ by using a flattening $\mathcal{C}'$ can only be done for those FOCTL*(Pr) properties that are preserved under such flattenings. Hence, it is important to determine which FOCTL*(Pr) properties are preserved when $\mathcal{C}$ and $\mathcal{C}'$ satisfy given relationships (see Theorem 1). The

most common relationship is precisely the equality of reachability sets (leading to the notion of post*-flattening). Let $\mathcal{C} = \langle \Sigma, Q, T \rangle$ be a PCS. The reachability sets from a configuration and from a set of Presburger definable configurations are defined as follows: $\mathrm{post}^*_{\mathcal{C}}(\langle q, \mathbf{a} \rangle) \stackrel{\text{def}}{=} \{\langle q', \mathbf{a}' \rangle : \langle q, \mathbf{a} \rangle \rightarrow^* \langle q', \mathbf{a}' \rangle \text{ in } S_{\mathcal{C}}\}$ and $\mathrm{post}^*_{\mathcal{C}}(q, \psi(\mathbf{x})) \stackrel{\text{def}}{=} \bigcup_{\langle q, \mathbf{a} \rangle \models \psi(\mathbf{x})} \mathrm{post}^*_{\mathcal{C}}(\langle q, \mathbf{a} \rangle)$.

Definition 7. *$\langle \mathcal{C}', q' \rangle$ is a f-**post*-flattening** (post*-flattening for short) of $\langle \mathcal{C}, q \rangle$ wrt $\psi(\mathbf{x})$ iff $\mathrm{post}^*_{\mathcal{C}}(q, \psi(\mathbf{x})) = f(\mathrm{post}^*_{\mathcal{C}'}(q', \psi(\mathbf{x})))$ and $\mathcal{C}'$ is a f-flattening of $\mathcal{C}$ (f is naturally extended to states of $\langle S_{\mathcal{C}}, \rightarrow_{\mathcal{C}} \rangle$).*

Even though it is undecidable whether a PCS has a post*-flattening [BFLS05, Theorem 4.9], we can decide if a PCS is a post*-flattening of another one.

Lemma 1. *Let $\langle \mathcal{C}', q' \rangle$ be an f-flattening of $\langle \mathcal{C}, q \rangle$ such that $\mathcal{C}'$ is an ACS. It is decidable to check whether $\langle \mathcal{C}', q' \rangle$ is a post*-flattening of $\langle \mathcal{C}, q \rangle$ wrt $\psi(\mathbf{x})$.*

Let $\mathcal{C} = \langle \Sigma, Q, T \rangle$ be a PCS. A *trace* for $\langle q, \mathbf{a} \rangle$ is a (possibly infinite) sequence of the form $\langle q_0, \mathbf{a_0} \rangle \langle q_1, \mathbf{a_1} \rangle \langle q_2, \mathbf{a_2} \rangle \ldots$ such that $\langle q_0, \mathbf{a_0} \rangle = \langle q, \mathbf{a} \rangle$, and for every i, $\langle q_i, \mathbf{a_i} \rangle \rightarrow \langle q_{i+1}, \mathbf{a_{i+1}} \rangle$ in $\langle S_{\mathcal{C}}, \rightarrow_{\mathcal{C}} \rangle$. The set of traces for $\langle q, \mathbf{a} \rangle$ in $\mathcal{C}$ is denoted by $\mathrm{traces}_{\mathcal{C}}(\langle q, \mathbf{a} \rangle)$. By extension, $\mathrm{traces}_{\mathcal{C}}(q, \psi(\mathbf{x})) \stackrel{\text{def}}{=} \bigcup_{\langle q, \mathbf{a} \rangle \models \psi(\mathbf{x})} \mathrm{traces}_{\mathcal{C}}(\langle q, \mathbf{a} \rangle)$.

Definition 8. *$\langle \mathcal{C}', q' \rangle$ is a f-**trace-flattening** (trace-flattening for short) of $\langle \mathcal{C}, q \rangle$ wrt $\psi(\mathbf{x})$ iff $\mathrm{traces}_{\mathcal{C}}(q, \psi(\mathbf{x})) = f(\mathrm{traces}_{\mathcal{C}'}(q', \psi(\mathbf{x})))$ and $\mathcal{C}'$ is a f-flattening of $\mathcal{C}$.*

We can decide if a PCS is a trace-flattening of another PCS as stated below.

Lemma 2. *Let $\langle \mathcal{C}', q' \rangle$ be an f-flattening of $\langle \mathcal{C}, q \rangle$ such that $\mathcal{C}'$ is an ACS. It is decidable to check whether $\langle \mathcal{C}', q' \rangle$ is a trace-flattening of $\langle \mathcal{C}, q \rangle$ wrt $\psi(\mathbf{x})$.*

Here is the more elaborate notion of flattenings but difficult to check.

Definition 9. *$\langle \mathcal{C}', q' \rangle$ is a f-**bisimulation-flattening** (bisimulation-flattening for short) of $\langle \mathcal{C}, q \rangle$ with respect to $\psi(\mathbf{x})$ iff $\mathcal{C}'$ is a f-flattening of $\mathcal{C}$ and for every $\mathbf{a}$ such that $\mathbf{a} \models \psi(\mathbf{x})$, $\langle \mathrm{post}^*_{\mathcal{C}'}(\langle q', \mathbf{a} \rangle), \rightarrow^{\mathbf{a}}_{\mathcal{C}'} \rangle$ where $\rightarrow^{\mathbf{a}}_{\mathcal{C}'}$ is the restriction of $\rightarrow_{\mathcal{C}'}$ to $\mathrm{post}^*_{\mathcal{C}}(\langle q, \mathbf{a} \rangle)$ is bisimilar to $\langle \mathrm{post}^*_{\mathcal{C}}(\langle q, \mathbf{a} \rangle), \rightarrow^{\mathbf{a}}_{\mathcal{C}} \rangle$.*

Lemma 3 below states a few easy results about flattenings and their hierarchy.

Lemma 3. *Let $\langle \mathcal{C}', q' \rangle$ be an f-flattening of $\langle \mathcal{C}, q \rangle$.*

(I) *For any $\psi(\mathbf{x})$, $f(\mathrm{post}^*_{\mathcal{C}'}(q', \psi(\mathbf{x}))) \subseteq \mathrm{post}^*_{\mathcal{C}}(q, \psi(\mathbf{x}))$.*
(II) *For any $\psi(\mathbf{x})$, $f(\mathrm{traces}_{\mathcal{C}'}(q', \psi(\mathbf{x}))) \subseteq \mathrm{traces}_{\mathcal{C}}(q, \psi(\mathbf{x}))$.*
(III) *Every bisimulation-flattening [resp. trace-flattening] is a trace-flattening [resp. post*-flattening].*

Based on standard properties of temporal logics, we provide below sufficient conditions to verify flattable PCS that are not necessarily ACS.

Theorem 1. *Let $\langle \mathcal{C}', q' \rangle$ be a post*-flattening [resp. trace-flattening, bisimulation-flattening] of the PCS $\langle \mathcal{C}, q \rangle$ wrt $\mathbf{a}$. Then, for every formula ϕ in the strict EF fragment [resp. the LTL fragment, FOCTL*(Pr)[n]], $\mathcal{C}', \langle q', \mathbf{a} \rangle \models \phi$ iff $\mathcal{C}, \langle q, \mathbf{a} \rangle \models \phi$.*

5 Model-Checking Flattable Counter Systems

Herein, we show decidability of model checking FOCTL*(Pr) over ACS and we propose a complete semi-algorithm for model checking FOLTL(Pr) formulae over trace-flattable PCS, extending what is done in [BFLS05] for post*-flattable PCS.

5.1 A FOCTL*(Pr) decision procedure to verify ACS

Throughout this section, let $\mathcal{C} = \langle \Sigma, Q, T \rangle$ be ACS of dimension n. Recall that all cycles in an ACS are simple cycles.

Definition 10. *A **control path** in $\mathcal{C}$ is any infinite path in $\mathcal{C}$. A **path segment** in $\mathcal{C}$ is a single transition $t \in T$ or a cycle in $\mathcal{C}$, and so is uniformly described as a finite sequence of control states. A **path schema** in $\mathcal{C}$ is a sequence $\langle \sigma_0, \ldots, \sigma_k \rangle$ of different path segments in $\mathcal{C}$ such that: (1) for every $0 \leq i \leq k-1$, the last control state of σ_i is the first control state of σ_{i+1}, (2) any path segment occurs at most once and (3) σ_k is a cycle. Cycles in a path schema that are not the final segment are called **interior cycles** of the schema.*

From now on we fix an enumeration $\lambda_1, \ldots, \lambda_M$ of all the cycles in $\mathcal{C}$ and assume that $M > 0$. Since an ACS is flat and has a finite number of control states, the following holds:

Proposition 4. *In every ACS $\mathcal{C}$, there is a finite number of path schemas.*

The number of path schemas is generally exponential in the size of the ACS. Hereafter we fix an enumeration $\langle 1, \ldots, P \rangle$ of the path schemas of $\mathcal{C}$. A path schema with at least one interior cycle corresponds to infinitely many different control paths, since any interior cycle in the schema may be repeated an arbitrary number of times on the control path. The number of repetitions of a given cycle in a control path is called the **cycle count** of that cycle. Thus, every control path is completely characterised by its underlying path schema and the cycle counts for its interior cycles. The next definition formalises this idea.

Definition 11. *Let the ACS $\mathcal{C}$ have $M > 0$ cycles and P path schemas. A **cycle count vector** $\mathbf{c}$ is a tuple $\langle c_1, \ldots, c_M \rangle \in \mathbb{N}^M$, where c_r represents the cycle count for the cycle λ_r. A **control path description** α is a pair $\alpha = \langle p, \mathbf{c} \rangle$ where $p \in \{1, \ldots, P\}$ denotes the path schema, $\mathbf{c}$ is the cycle count vector for the control path being described, $c_i > 0$ for every interior cycle λ_i and $c_i = 0$ for any cycle λ_i in $\mathcal{C}$ which is not interior in the path schema p. Hereafter a control path description, may be written as $\langle p, c_1, \ldots, c_M \rangle$. We write α_0 for the path schema associated with control path description α.*

The following is immediate from the flatness condition on ACS.

Proposition 5. *For every control path in $\mathcal{C}$ there is a unique control path description.*

Without risk of confusion, we identify every control path with its description.

Every configuration path is uniquely described by the pair $\langle \alpha, \langle q, \mathbf{a} \rangle \rangle$ where α is its control path and $\langle q, \mathbf{a} \rangle$ is the initial configuration. Conversely, due to the functionality of $\mathcal{C}$, every such pair $\langle \alpha, \langle q, \mathbf{a} \rangle \rangle$ describes a unique path in the configuration graph starting at $\langle q, \mathbf{a} \rangle$, and progressing according to the transitions of the control path α. Note, however, that such a path may terminate and therefore not be considered as a configuration path. There exists a Presburger formula that exactly describes the configuration path associated with a control path and initial configuration as stated below.

Theorem 2. *Given the ACS $\mathcal{C}$ of dimension n with $M > 0$ cycles, one can compute a Presburger formula $PathConfig_{\mathcal{C}}(\xi, \mathbf{x}, i, \mathbf{y})$ such that for all $\alpha \in \mathbb{N}^{M+1}$, $\mathbf{a} \in \mathbb{N}^{n+1}$, $m \in \mathbb{N}$ and $\mathbf{b} \in \mathbb{N}^{n+1}$ $\alpha, \mathbf{a}, m, \mathbf{b} \models PathConfig_{\mathcal{C}}(\xi, \mathbf{x}, i, \mathbf{y})$ iff α is a valid control path description and the m^{th} configuration of the configuration path $\langle \alpha, \mathbf{a} \rangle$ is $\mathbf{b}$.*

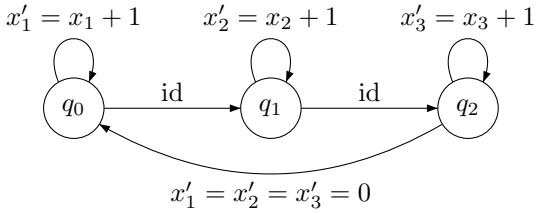
Now we are ready to show that model-checking FOCTL*(Pr)[n] can be reduced to satisfiability in Presburger arithmetic.

Theorem 3. *Given an ACS $\mathcal{C}$ of dimension n with Presburger transition system $\langle S_{\mathcal{C}}, \rightarrow_{\mathcal{C}} \rangle$, for every FOCTL*(Pr)[$n$] formula φ, one can compute a Presburger formula $\psi(\mathbf{x})$ such that for every $\langle q, \mathbf{a} \rangle \in S_{\mathcal{C}}$, $\langle q, \mathbf{a} \rangle \models \psi(\mathbf{x})$ iff $\mathcal{C}, \langle q, \mathbf{a} \rangle \models \varphi$.*

For a fixed ACS, the size of $\psi(\mathbf{x})$ is linear in the size of φ.

Theorem 4. *The two problems in Section 3.2 are decidable.*

Theorem 4 can be easily extended to allow past-time operators such as 'previous' $\mathtt{X}^{-1}$ and 'since' $\mathtt{S}$. By contrast, we state below an undecidability result for a fixed PCS that is almost an ACS. We present a PCS $\mathcal{C}_u$ that is obtained from an ACS by only adding a reset transition while preserving the Presburger counter acceleration property and functionality (see below).



$\mathcal{C}_u$ is of dimension 4 with counters x_0, x_1, x_2 and x_3 and x_0 is the counter related to the control state. "id" denotes the identity function on the counters x_1, x_2 and x_3.

Theorem 5. *Local model-checking on $\mathcal{C}_u$ with FOLTL*(Pr)[3] is Σ^1_1-hard.*

Observe that $\mathcal{C}_u$ admits a post*-flattening with an ACS and therefore the strict EF fragment has a decidable local model-checking problem for $\mathcal{C}_u$.

5.2 Model-Checking of Trace-Flattable Counter Systems

Suppose we have a functional PCS $\mathcal{C}$ with the Presburger counting acceleration property. Typically, $\mathcal{C}$ can be a linear PCS with finite monoid. Let $\langle q, \mathbf{a}\rangle$ be a configuration for which we want to check a FOLTL(Pr) formula ϕ. We propose below the basis of a semi-algorithm `model-check` to verify whether $\mathcal{C}, \langle q, \mathbf{a}\rangle \models \phi$.

procedure `model-check`$(\mathcal{C}, \langle q, \mathbf{a}\rangle, \phi)$

1. $found := false$;
2. **while** not $found$ **do**
   (a) Choose fairly a flattening $\langle \mathcal{C}', q'\rangle$ of $\langle \mathcal{C}, q\rangle$;
   (b) **if** $\langle \mathcal{C}', q'\rangle$ is a trace-flattening of $\langle \mathcal{C}, q\rangle$ **then** $found := true$;
3. **return** $\mathcal{C}', \langle q', \mathbf{a}\rangle \models \phi$.

To become efficient, the semi-algorithm has to be refined in order to obtain an efficient enumeration of the flat PCS as that is done with the tool FAST. As a first step, heuristics implemented in FAST can be used, see e.g. [BFLS05]. The semi-algorithm `model-check` extends the underlying FAST algorithm [BFLS05] to trace-flattable Presburger counter systems and LTL temporal properties which paves the way to design the new generation of the tool. Using previous results shown in the paper, we can establish the following key result of the paper.

Theorem 6. *(I)* `model-check`*$(\mathcal{C}, \langle q, \mathbf{a}\rangle, \phi)$ terminates iff $\mathcal{C}$ has a trace-flattening wrt to $\langle q, \mathbf{a}\rangle$. (II) When* `model-check`*$(\mathcal{C}, \langle q, \mathbf{a}\rangle, \phi)$ terminates, it returns whether $\mathcal{C}, \langle q, \mathbf{a}\rangle \models \phi$ holds true.*

Proof. It is sufficient to observe the following facts:

- Checking whether $\langle \mathcal{C}', q'\rangle$ is a flattening of $\langle \mathcal{C}, q\rangle$ can be done in exponential-time.
- Checking whether $\mathcal{C}'$ is an ACS is easy since $\mathcal{C}$ has the Presburger counting acceleration property and it is functional. Hence $\mathcal{C}$ is an ACS and one can compute effectively the Presburger formulae related to cycles.
- Checking whether $\langle \mathcal{C}', q'\rangle$ is a trace-flattening of $\langle \mathcal{C}, q\rangle$ is decidable as a consequence of Lemma 2.
- Checking whether $\mathcal{C}', \langle q', \mathbf{a}\rangle \models \phi$ is decidable by Theorem 4.
- Finally, $\mathcal{C}', \langle q', \mathbf{a}\rangle \models \phi$ iff $\mathcal{C}, \langle q, \mathbf{a}\rangle \models \phi$ by Theorem 1. □

We do not know yet how to extend the above complete semi-algorithm to deal with bisimulation-flattening. Indeed, in order to have a decision procedure for the step (3) with bisimulation, we would need decidability of some kind of modal mu-calculus over ACS, which is open so far.

5.3 Decidable Extension with CQDD Patterns

We present below an extension of FOCTL*(Pr)[n] for which model-checking over ACS can be also encoded into Presburger satisfiability. In a seminal paper, Wolper extends LTL to an extended temporal logic that has the same power as Büchi automata [Wol83]. In this section, we extend the set of path formulae from FOCTL*(Pr)[n] by allowing temporal operators defined by another class of language acceptors, namely the CQDD (*constrained queue-content decision diagrams*) [BH99]. This formalism has been introduced for representing symbolically infinite sets of configurations in FIFO automata. Our use of CQDD is different and non-regular languages can be defined with CQDD. Moreover, the model-checking problem for LTL augmented with operators defined from CQDD is undecidable [Dem06] unlike the extension with regular languages [Wol83]. By contrast, we show that the model-checking problem for FOCTL*(Pr)[n] extended with CQDD-based operators is decidable over ACS. Regain of decidability is due to the flatness restriction in CQDD. Hence, we show evidence in this section that we can take advantage of flatness in models *and* in formulae. A **CQDD** is a structure $\mathcal{A} = \langle \Sigma, S, S_0, E, l, \psi(y_1, \ldots, y_m), F\rangle$ such that:

- Σ is a finite alphabet and S is a finite set of states,
- $S_0 \subseteq S$ [resp. $F \subseteq S$] is the set of initial [resp. final] states,
- $E \subseteq S \times \Sigma \times S$ is a set of transitions of cardinality m and $\langle S, E\rangle$ is flat,
- l is a bijection from E to $\{1, \ldots, m\}$ and $\psi(y_1, \ldots, y_m)$ is a Presburger formula.

An accepting run is a sequence $q_0 \xrightarrow{a_0} q_1 \xrightarrow{a_1} q_2 \ldots \xrightarrow{a_k-1} q_k$ such that $q_0 \in S_0$, $q_k \in F$, for every $i \in \{0, \ldots, k-1\}$, $\langle q_i, a_i, q_{i+1}\rangle \in E$, and $n_1, \ldots, n_m \models \psi(y_1, \ldots, y_m)$ in Presburger arithmetic, where each n_i is the number of occurrences of the transition $l^{-1}(i)$ in the sequence. The word $\sigma \in \Sigma^*$ is accepted by the accepting run $q_0 \xrightarrow{a_0} q_1 \xrightarrow{a_1} q_2 \ldots \xrightarrow{a_{k-1}} q_k$ whenever $\sigma = a_0 a_1 a_2 \ldots a_{k-1}$. The word σ is also said to be accepted by the automaton $\mathcal{A}$. We write L($\mathcal{A}$) to denote the set of words accepted by $\mathcal{A}$.

Let $\mathcal{A} = \langle \Sigma, S, S_0, E, l, \psi(y_1, \ldots, y_m), F\rangle$ be a CQDD with the letters from Σ linearly ordered: $a_1 < \ldots < a_k$. The extension EFOCTL*(Pr)[n] of the logic FOCTL*(Pr)[n] consists in considering formulae of the form $\mathcal{A}(\phi_1, \ldots, \phi_k)$ defined as follows: $\pi, i \models \mathcal{A}(\phi_1, \ldots, \phi_k)$ iff: either $\epsilon \in$ L($\mathcal{A}$), or there is a finite word $a_{i_1} a_{i_2} \ldots a_{i_n} \in$ L($\mathcal{A}$) such that for every $1 \leq j \leq n$, $\pi, i + (j-1) \models \phi_{i_j}$. For instance, in EFOCTL*(Pr)[n] we can state that there is a path and some $n \neq 0$ such that ϕ_1 holds true at the n first positions, then ϕ_2 holds true at the n next positions and then neither ϕ_1 nor ϕ_2 holds true forever. It is known that ETL is more expressive that LTL [Wol83] and this result can be lifted between FOCTL*(Pr)[n] and EFOCTL*(Pr)[n]. Theorem 3 can be extended by allowing CQDD-based operators.

Theorem 7. *Given an ACS $\mathcal{C}$ of dimension n with Presburger transition system $\langle S_\mathcal{C}, \rightarrow_\mathcal{C}\rangle$, for every EFOCTL*(Pr)[$n$] formula φ, one can compute a Presburger formula $\psi(\mathbf{x})$ such that for every $\langle q, \mathbf{a}\rangle \in S_\mathcal{C}$, $\langle q, \mathbf{a}\rangle \models \psi(\mathbf{x})$ iff $\mathcal{C}, \langle q, \mathbf{a}\rangle \models \varphi$.*

As a corollary, local model-checking problem for EFOCTL*(Pr)$[n]$ over ACS is decidable.

6 Concluding Remarks

We have designed a complete semi-algorithm to verify first-order LTL properties over trace-flattable counter systems, extending the underlying semi-algorithm to verify reachability questions over post*-flattable systems in the tool FAST. We expect a smooth extension of FAST [BFLS05] to deal with trace-flattable systems. This result takes strongly advantage of the decidability of model-checking FOCTL*(Pr) over admissible counter systems, a new result we establish in the paper. Hence, we have improved the decidability boundary for model-checking ACS with CTL*-like languages. The decidability of model-checking question is open when adding fixed-point operators (Presburger mu-calculus) or monadic second-order quantification over ACS. Another direction for further work is to analyze and extend further the class of ACS. For instance, giving up the functionality assumption on transitions that do not belong to a cycle preserves decidability, while it is open whether giving up the full functionality assumption still preserves decidability in the absence of first-order quantification. Finally, we plan to verify experimentally which post*-flattable case studies [BFLS05] are indeed trace-flattable.

References

[BCMS01] O. Burkart, D. Caucal, F. Moller, and B. Steffen. Verification of infinite structures. In *Handbook of Process Algebra*, pages 545–623. Elsevier, 2001.

[BDR03] V. Bruyère, E. Dall'Olio, and J.F. Raskin. Durations, parametric model-checking in timed automata with presburger arithmetic. In *STACS'03*, volume 2607 of *LNCS*, pages 687–698. Springer, 2003.

[BEH95] A. Bouajjani, R. Echahed, and P. Habermehl. On the verification problem of nonregular properties for nonregular processes. In *LICS'95*, pages 123–133, 1995.

[BEM97] A. Bouajjani, J. Esparza, and O. Maler. Reachability analysis of pushdown automata: Application to model checking. In *CONCUR'97*, volume 1243 of *LNCS*, pages 135–150. Springer, 1997.

[BFL04] S. Bardin, A. Finkel, and J. Leroux. FASTer acceleration of counter automata in practice. In *TACAS'04*, volume 2988 of *LNCS*, pages 576–590. Springer, March 2004.

[BFLP03] S. Bardin, A. Finkel, J. Leroux, and L. Petrucci. FAST: Fast Acceleration of Symbolic Transition systems. In *CAV'03*, volume 2725 of *LNCS*, pages 118–121. Springer, 2003.

[BFLS05] S. Bardin, A. Finkel, J. Leroux, and P. Schnoebelen. Flat acceleration in symbolic model checking. In *ATVA'05*, volume 3707 of *LNCS*, pages 474–488. Springer, 2005.

[BGP97] T. Bultan, R. Gerber, and W. Pugh. Symbolic model checking of infinite state systems using Presburger arithmetic. In *CAV'97*, volume 1254 of *LNCS*, pages 400–411. Springer, 1997.

[BH99] A. Bouajjani and P. Habermehl. Symbolic reachability analysis of FIFO-channel systems with nonregular sets of configurations. *TCS*, 221(1–2):211–250, 1999.

[Boi98] B. Boigelot. *Symbolic methods for exploring infinite state spaces*. PhD thesis, Université de Liège, 1998.

[CC00] H. Comon and V. Cortier. Flatness is not a weakness. In *CSL'00*, volume 1862 of *LNCS*, pages 262–276. Springer, 2000.

[CJ98] H. Comon and Y. Jurski. Multiple counters automata, safety analysis and Presburger analysis. In *CAV'98*, volume 1427 of *LNCS*, pages 268–279. Springer, 1998.

[Cor02] V. Cortier. About the decision of reachability for register machines. *Theoretical Informatics and Applications*, 36(4):341–358, 2002.

[Dem06] S. Demri. Temporal logics. Lecture notes for MPRI, 2005/2006. `www.lsv.ens-cachan.fr/~demri/`.

[DPK03] Z. Dang, P. San Pietro, and R. Kemmerer. Presburger liveness verification of discrete timed automata. *TCS*, 299:413–438, 2003.

[EFM99] J. Esparza, A. Finkel, and R. Mayr. On the verification of broadcast protocols. In *LICS'99*, pages 352–359, 1999.

[FL02] A. Finkel and J. Leroux. How to compose Presburger accelerations: Applications to broadcast protocols. In *FST&TCS'02*, volume 2256 of *LNCS*, pages 145–156. Springer, 2002.

[FO97] L. Fribourg and H. Olsén. Proving safety properties of infinite state systems by compilation into presburger arithmetic. In *CONCUR'97*, volume 1243 of *LNCS*, pages 213–227. Springer, 1997.

[FWW97] A. Finkel, B. Willems, and P. Wolper. A direct symbolic approach to model checking pushdown systems (extended abstract). In *INFINITY'97*, volume 9 of *ENTCS*. Elsevier Science, 1997.

[Iba78] O. Ibarra. Reversal-bounded multicounter machines and their decision problems. *JACM*, 25(1):116–133, 1978.

[ISD+00] O. Ibarra, J. Su, Z. Dang, T. Bultan, and A. Kemmerer. Counter machines: Decidable properties and applications to verification problems. In *MFCS'00*, volume 1893 of *LNCS*, pages 426–435. Springer, 2000.

[Lag85] J. Lagarias. The $3x + 1$ problem and its generalizations. *The American Mathematical Monthly*, 92(1):3–23, 1985.

[Ler03] J. Leroux. *Algorithmique de la vérification des systèmes à compteurs. Approximation et accélération. Implémentation de l'outil FAST*. PhD thesis, ENS de Cachan, France, 2003.

[Ler06] J. Leroux. Regular acceleration for number decision diagrams. Technical Report 1385-06, LABRI, January 2006.

[LS05] J. Leroux and G. Sutre. Flat counter systems are everywhere! In *ATVA'05*, volume 3707 of *LNCS*, pages 489–503. Springer, 2005.

[Min67] M. Minsky. *Computation, Finite and Infinite Machines*. Prentice Hall, 1967.

[SS04] T. Schuele and K. Schneider. Global vs. local model checking: A comparison of verification techniques for infinite state systems. In *SEFM'04*, pages 67–76. IEEE, 2004.

[Wal01] I. Walukiewicz. Pushdown processes: games and model-checking. *I & C*, 164(2):234–263, 2001.

[Wol83] P. Wolper. Temporal logic can be more expressive. *I & C*, 56:72–99, 1983.